

Self-Driving Car Engineer Nanodegree

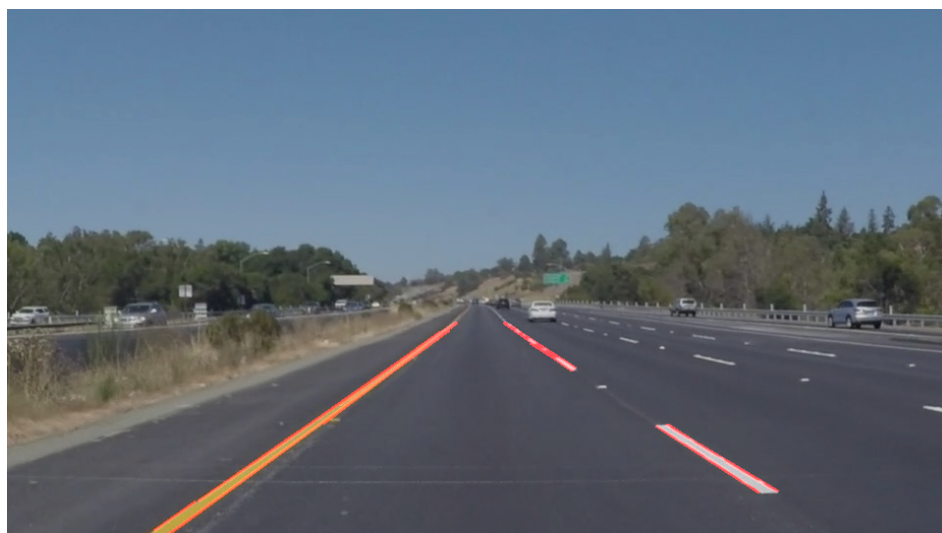
Project: Finding Lane Lines on the Road

The goals / steps of this project are the following:

1. Make a pipeline that finds lane lines on the road
2. Reflect on your work in a written report

Reflection

1. My pipeline consisted of 9 steps. First, I made a separate function `pipe_line(image)` which takes an image as an argument to get the desired output.
2. In order to draw a single line on the left and right lanes, first I used my learnings for module to detect line segments of lane. To achieve this I used helper functions provided in the project.
3. First converted image to grayscale and then used `gaussian_blur()` helper function,
4. On `gaussian_blur()` output I used `canny()` function to find out edges.
5. After canny transform the output was used to generate `region_of_interest()` for the image.
6. And then applied `hough_lines()` to get an image with hough lines drawn.
7. Finally I used `weighted_img()` helper function to get line segments of the lane. Following is one of the image in which line segment is detected:



8. After detecting the line segments, to draw a single line I used the hint provided in `draw_line()` and modified it, by separating line segments by their slope $((y_2 - y_1) / (x_2 - x_1))$ to decide which segments are part of the left line vs. the right line. If slope is positive I added them to left array and if negative then added them to right array.
9. After separating them I calculated the average position of each line and then used `cv2.line` to draw the pipeline.



Shortcomings

1. Tested on only test images provided.
2. All images are of day, could show different behaviour for the images taken in night or very low light.
3. Dimensions of all Images are same hence some values are hardcoded.

Improvements

1. Region of interest can be more optimised.